

✦ **Data de Entrega:** 22 de setembro

✦ **Valor:** 20 vistos, distribuídos da seguinte forma:

- **10 vistos** – Relatório (entrega obrigatória via Moodle)
- **10 vistos** – Código (no Drive, organizado na sua pasta correspondente)

Entrega via drive compartilhado, os códigos:

- Código fonte;
- Arquivos de entrada;
- Arquivos de tempo;
- Arquivos de saída

Entrega via moodle

- Relatório completo do trabalho prático;

Ex de organização para cada algoritmo, no caso abaixo o exemplo do Insertion Sort:

Implementar o Algoritmo Insertion Sort (visto em sala de aula) usando c ou c++

1. Programa fonte contendo o código C;
2. Arquivos de entradas crescente, random e decrescente para todos os algoritmos;

C:\arquivodeentrada\entradacrescente10.txt
C:\arquivodeentrada\entradacrescente100.txt
C:\arquivodeentrada\entradacrescente1000.txt
C:\arquivodeentrada\entradacrescente1000.txt
C:\arquivodeentrada\entradacrescente10000.txt
C:\arquivodeentrada\entradacrescente100000.txt

C:\arquivodeentrada\entradadecrescente10.txt
C:\arquivodeentrada\entradadecrescente100.txt
C:\arquivodeentrada\entradadecrescente1000.txt
C:\arquivodeentrada\entradadecrescente1000.txt
C:\arquivodeentrada\entradadecrescente10000.txt
C:\arquivodeentrada\entradadecrescente100000.txt

C:\arquivodeentrada\entradarandom10.txt
C:\arquivodeentrada\entradarandom100.txt
C:\arquivodeentrada\entradarandom1000.txt
C:\arquivodeentrada\entradarandom1000.txt
C:\arquivodeentrada\entradarandom10000.txt
C:\arquivodeentrada\entradarandom100000.txt

Sugestão: gerar as entradas; carregar as entradas no vetor para fazer a Ordenação, em seguida gravar o tempo e a saída ordenada em arquivo. Ou seja, mesclar arquivo de vetor.

FASES:

Implemente em linguagem C o algoritmo de ordenação Insertion sort_versao1(algoritmo nominal apresentado em sala de aula),

Implemente um Menu na aplicação para melhor interação com o usuário. Esse menu poderá conter (sugestão):

1 As opções de InsertionSort - ex.(1: INSERTIONSORT EM VETOR,

A forma em que serão geradas as instâncias:

'r' para números randômicos.

'c' para números crescentes.

'd' para números decrescentes.

3 :Opção de sair

Gere instâncias de tamanhos: 10, 100, 1.000, 10.000 e 100.000 e 1.000.000 para todas as formas (randômico, crescente e decrescente), e execute o algoritmo implementado.

IMPORTANTE: Na parte de arquivo, gere:

1- Um arquivo de entrada contendo as instâncias geradas(os números gerados para a ordenação).

2- Um arquivo contendo as instâncias ordenadas(É preciso salvar nesse arquivo de saída, os números ordenados pelo algoritmo).

3- Um arquivo contendo o tempo gasto pela ordenação.

Os arquivos tanto os de entrada como os de saída devem ter na primeira linha o tamanho da instância e o restante dos dados, um por linha, como é indicado na Figura 1

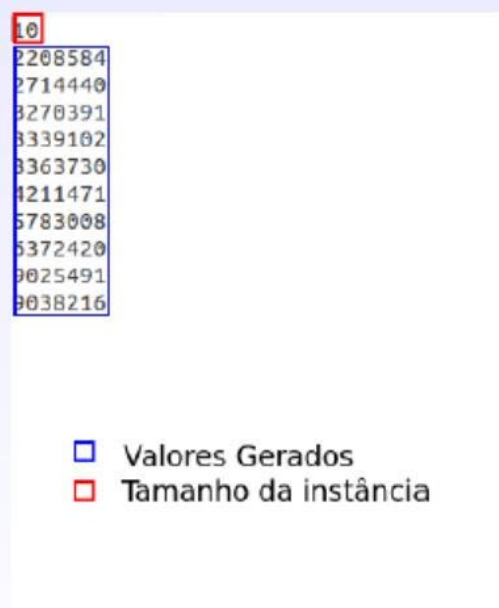


Figura: Exemplo de arquivo de entrada.

Os arquivos tanto os de entrada como os de tempo devem ter na primeira linha o tamanho da instância e o restante dos dados, um por linha, como é indicado na Figura acima .

Sumário do relatório:

Sumário

1 INTRODUÇÃO

2 ALGORITMO *INSERTION SORT*

3 ANÁLISE E COMPLEXIDADE DO ALGORITMO

4 TABELA/GRÁFICO DO ALGORITMO

5 REFERÊNCIAS BIBLIOGRÁFICAS